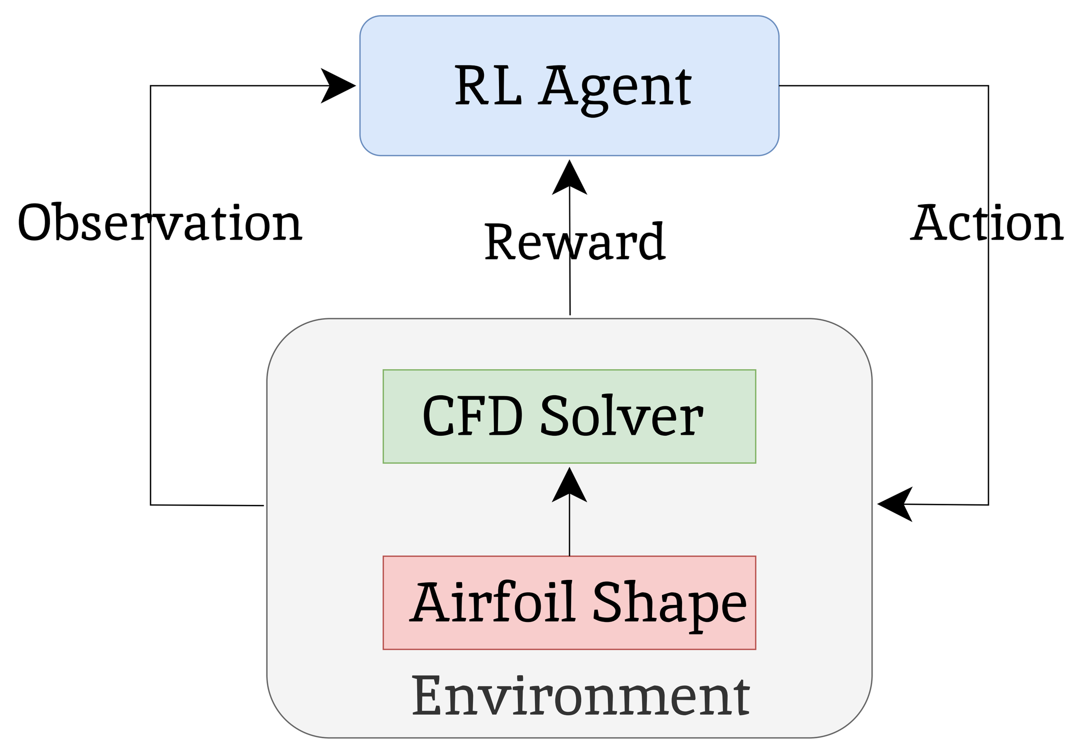
Robust Shape Optimization using Neural Injective Geometry networks

Atharva Aalok
Aeronautics and Astronautics
Stanford University

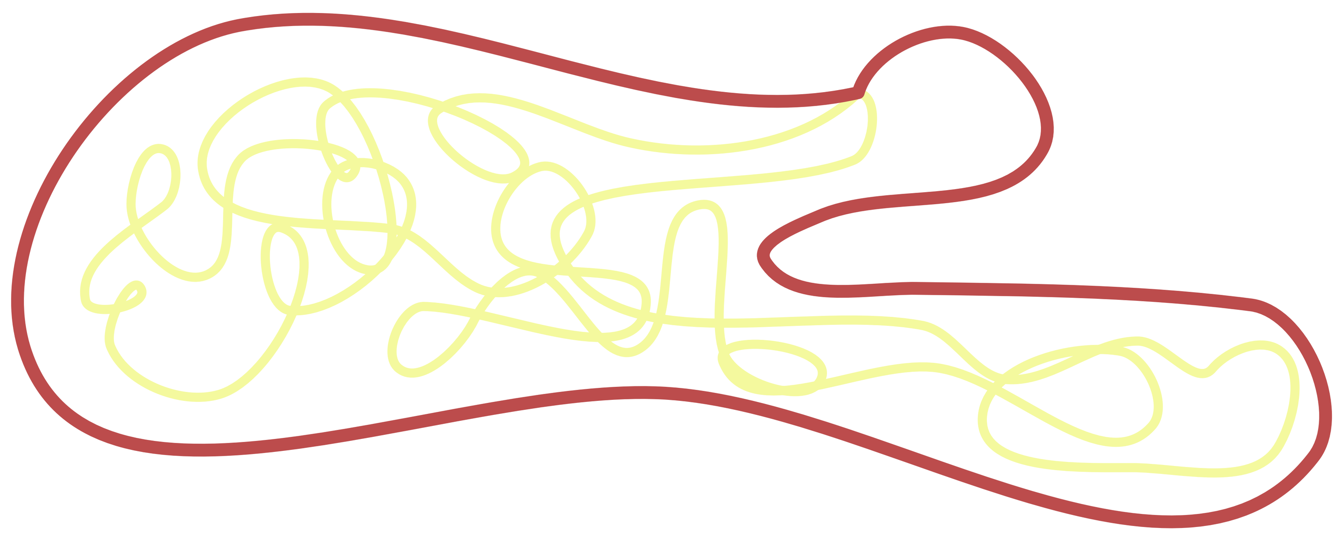
Aerodynamic Shape Optimization

We use Deep Reinforcement Learning to optimize airfoils by intelligently searching the design space. Airfoils are aircraft wing cross-section shapes and dictate aerodynamic properties such as lift and drag. We design optimal shapes with the highest lift-to-drag ratio.



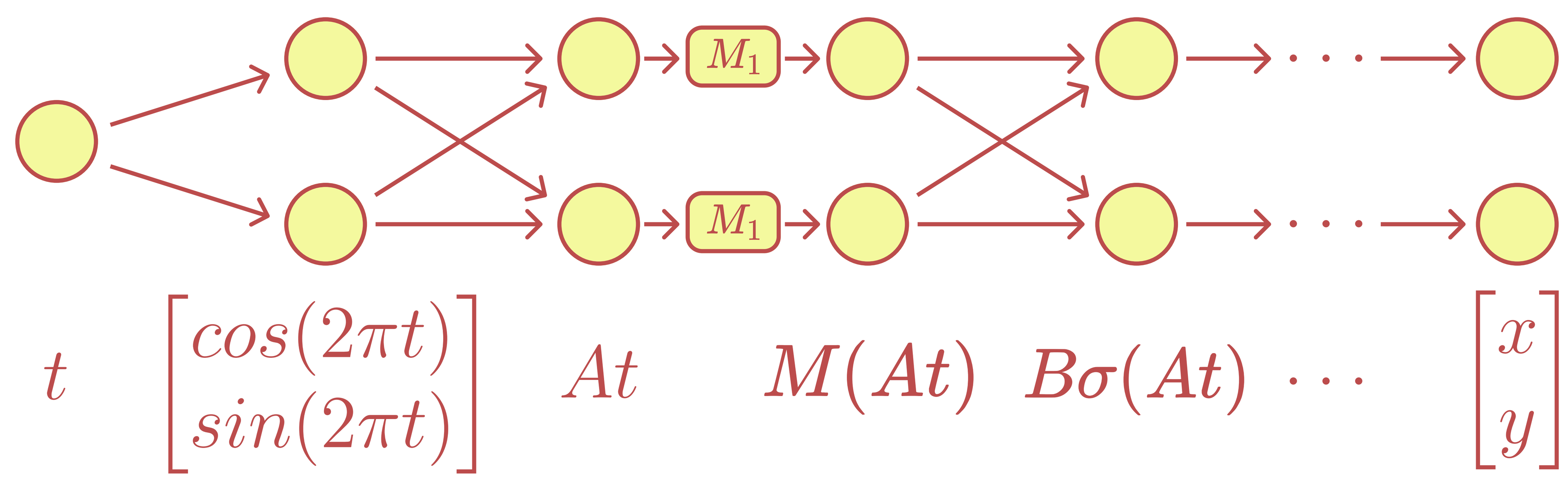
The Problem: Self-Intersection

Until now DeepRL based shape optimization [2, 3] has used specifically constrained shape representations. General methods dealing with arbitrary shaped curves have not yet worked because methods like splines lead to self-intersecting geometry. One issue especially problematic for RL is Design Space Blow-Up. Imagine flow going left → right over the shape below. It never interacts with the internal representation making it wasteful. This leads to an agent searching through a design space bigger than it should.



NIGnets and the RL Pipeline

To tackle the self-intersection problem we will use Neural Injective Geometry networks (NIGnets). We developed this to give a hard guarantee on representing only non-self-intersecting geometry. The core architecture looks as follows:

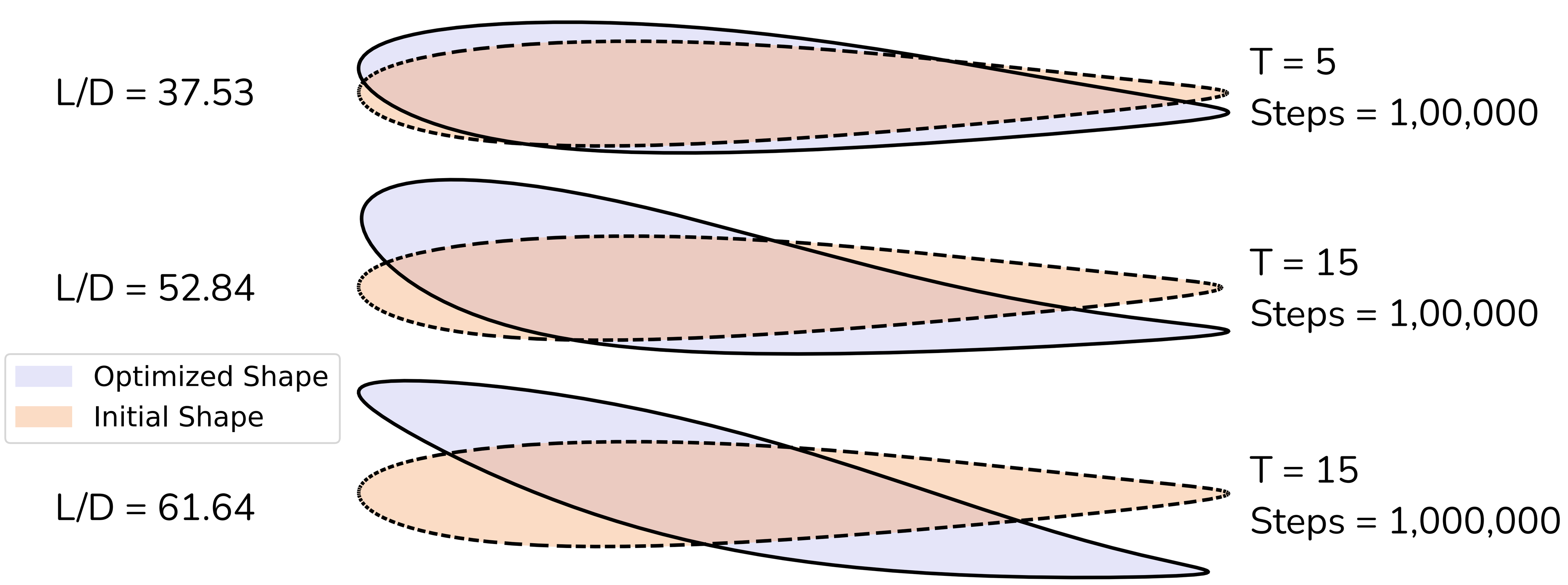


Our DeepRL based shape optimization pipeline works as follows:

1. Represent state (shape) using NIGnet parameters ϕ.
2. An action corresponds to outputting $\Delta\phi$ that perturbs the NIGnet parameters $\phi' = \phi + \Delta\phi$.
3. A reward equal to the L/D ratio is produced for the shape that the NIGnet ϕ' represents.

Results

We instantiate the RL problem as one with a fixed time horizon T. At each stage the agent can produce a network parameter perturbation in the clipped interval [-σ, σ] and we fix σ = 0.01. We use PPO with an MLP policy with 2 hidden layers of width 64. We perform training runs for progressively increasing T and total training Steps and observe performance. We use XFoil to compute L/D and use that as our reward. On non-convergence we give a reward of -50.



Discussion

1. The most interesting observation is that for a small number of training steps and for a short horizon an increase in L/D comes almost completely from an increase in the angle of attack with barely any shape change.
2. On increasing the time horizon we observe a further increase in angle of attack and also start seeing some shape morphing.
3. If we also crank up the total training Steps then we witness significant shape morphing on top of a high angle of attack.
4. We also experimented with a larger time horizon of T = 20 but in that case the final few shapes are so morphed that they do not converge in XFoil and negative rewards hinder good training.

Limitations and Future Work

1. Using a more robust CFD solver. For our solver we use Xfoil which is fast but not very robust and on non-convergence in the flow we end up giving a negative reward which is not very meaningful.
2. Use goal-conditioning style methods to make the design more diverse than just maximizing L/D. We can then train the agent to design shapes given a target L/D ratio.

References

1. Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).
2. Viquerat, Jonathan, et al. "Direct shape optimization through deep reinforcement learning." Journal of Computational Physics 428 (2021): 110080.
3. Dussauge, Thomas P., et al. "A reinforcement learning approach to airfoil shape optimization." Scientific Reports 13.1 (2023): 9753.